

REMARKS

This Amendment, submitted in response to the Office Action dated April 19, 2007, is believed to be fully responsive to each point of rejection raised therein. Accordingly, favorable reconsideration on the merits is respectfully requested.

Claims 1-19 are pending in the present application.

I. Claim Rejections under 35 U.S.C. § 112

Claim 17 stands rejected under 35 U.S.C. § 112, first paragraph, as allegedly containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention.

Specifically, the Examiner asserts that the recitation “wherein the grouping of all of said data update requests into the plurality of blocks is performed at a same time” is not disclosed in the specification.

However, Applicant submits that as illustrated, for example, in Applicant’s Fig. 1, the data update requests are accumulated in the update data file 12 in the order of processing by a bookkeeping system 16. See for example, Applicant’s specification at page 7, lines 16-19. Further, each block in a package is filled with data update requests before the next block in the sequence is filled with data update requests to ensure more efficient processing by the respective data processing part. See Applicant’s specification page 9, lines 16-19.

Therefore, Applicant submits that the Applicant’s specification and figures clearly discloses “the grouping of all of said data update requests into the plurality of blocks is performed at a same time,” as claimed. Further, Applicant submits that this aspect of the claim is not disclosed in Pupipeddi which clearly discloses the addition of data requests at different points

in time and the subsequent reordering of queues according to the time of the addition of a new data request. See Pudipeddi Figs. 8A-8B.

For at least the above reasons, Applicant submits that the 35 U.S.C. § 112, first paragraph rejection of claim 17 should be withdrawn.

II. Claim Rejections under 35 U.S.C. § 103

Claims 1-10 and 16-19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pudipeddi et al.(U.S. Pub. No. 2005/0165820; hereinafter “Pudipeddi”) in view of Goudie et al. (U.S. Pub. No. 2002/0156764; hereinafter “Goudie”).

Claim 1 recites:

“grouping **all of said data update requests** which is followed by updating of the corresponding data into a predetermined plurality of blocks for execution by a data processor, the data update requests within each of said blocks and from one of said blocks to a next one of said blocks being arranged in an order that said data update requests need to be executed to yield a proper data result... and

then said data processor processing said data update requests within said one block in said order, and then said data processor processing said data update requests within said next block in said order.”

On page 9 of the Office Action, the Examiner asserts that the claims do not clearly disclose what the word “all” involves. The Examiner asserts that it is unclear whether “all” refers to all incoming requests or all the requests residing in the memory.

Applicant submits that claim 1 recites:

“A method for processing a multiplicity of data update requests made by a customer, said method comprising the steps of:

grouping all of said data update requests which is followed by updating of the corresponding data into a predetermined plurality of blocks for execution by a data

processor...”

Therefore, Applicant submits that based on the claim recitation, all of said data update requests refers to the multiplicity of data update requests made by a user. As previously submitted, Pudipeddi does not teach this aspect of the claim.

The Examiner asserts that queues 402 of Pudipeddi teach the grouping of all data requests into a plurality of blocks. The queues 402 of Pudipeddi include items 406 which are requests to retrieve data stored in for example, an optical disk. However, the queues 402 (cited by the examiner for teaching the claimed blocks of data update requests) do not contain all of the data update requests (made by a customer). In particular, Pudipeddi is not concerned with grouping all requests for data. As illustrated in Figs. 8A-8E of Pudipeddi, requests for data stored in a storage media are grouped into two monotonically increasing sequences. See para. 96. The reason Pudipeddi uses two monotonically increasing sequences is to address the situation where a request for a data object is dynamically queued during the reading of a medium. Therefore, in the event the location of a request has already passed its location on a first sequence, the request can then be inserted in a location in the second sequence. See para. 88. Thus, Pudipeddi discloses the insertion of individual requests which are not grouped in a block (queues 402 as cited by the Examiner). Since Pudipeddi is concerned with the dynamic insertion of requests, Pudipeddi does not teach grouping all of said data update requests (made by a customer) into a plurality of blocks.

Further, Pudipeddi does not disclose “grouping all of said data update requests which is followed by updating of the corresponding data into a predetermined plurality of blocks” as claimed.

Claim 1 also recites “each of said blocks having approximately **a same capacity for said**

data update requests, said **capacity corresponding to a number of said data update requests** which said data processor is adapted to efficiently process in order before processing said data update requests in the next one of said blocks.” The Examiner concedes that Pudipeddi does not teach this aspect of the claim cites Goudie to cure the deficiency.

In particular, the Examiner asserts that Goudie teaches a method of managing data in which the plurality of queues use a memory of the same size, citing para. 23 of Goudie in support. The aspect of Goudie cited by the Examiner discloses that all of the element queues in a grouped queue use memory blocks of the same size. However, although the blocks in a queue are the same size, there is no teaching or suggestion that the queues (blocks of update requests as asserted by the Examiner) of Goudie have the same capacity. In addition, although the memory blocks of Goudie are the same size, there is no teaching or suggestion that the capacity of the memory blocks correspond to a number of data update requests. Further, the size of the memory blocks of Goudie have no relationship with a number of data update requests. In particular, there is no teaching or suggestion that the blocks of Goudie include data update requests.

In response to the Applicant’s arguments, on page 10 of the Office Action, the Examiner asserts that one skilled in the art would know that capacity corresponds to size and that storing requests will also consume memory space and therefore, there is a limit of how many requests actually can be recorded. Therefore, capacity (interpreted as size) clearly corresponds to the number of requests.

However, Applicant submits that assuming one of skill in the art would equate size with capacity, one of skill in the art would not assume that the size of memory blocks in Goudie would have a size (capacity as cited by the Examiner) **corresponding to a number of said data update requests** since Goudie is not at all concerned with data update requests. Any suggestion

by the Examiner otherwise is clearly a result of impermissible hindsight upon viewing the Applicant's disclosure.

The Examiner asserts it would have been obvious to use Pudipeddi's requests processing method and system for the update information because selecting particular type of requests to be processed by the processing unit is just an intended use and is not different from processing any other type of request. Further, the Examiner asserts that it would have been obvious to use the queues with the same capacity as taught by Goudie because predefining the queue's size is a design choice and that setting all the blocks to the same size is simpler than adjusting the individual size of the particular queue or block.

Contrary to the Examiner's assertions, the combination of Goudie with Pudipeddi is not obvious. Pudipeddi is not at all concerned with a data update. Pudipeddi is concerned with the recall of stored backup data. If Pudipeddi were modified as suggested by the Examiner to include data update requests, this would defeat the stored data retrieval of Pudipeddi. In particular, if update data requests were retrieved, at no point would stored backup data be recalled.

Further, Pudipeddi is directed to dynamic adjustment of queues. See Figs. 8A-8E. Therefore, Pudipeddi is directed to adjusting the size of a queue. Consequently, it would be contrary to the operation of Pudipeddi that the queue sizes be of a same capacity, further evidencing that the combination of Goudie with Pudipeddi is not obvious. MPEP 2143.01 "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959)." Therefore, it would not be obvious that the type of data request of Pudipeddi

would include a data update request because that would be completely contrary to the recall of stored backup data as taught in Pudipeddi.

The Examiner then goes on to say on page 11 of the Office Action that the recall of stored backup data disclosed in Pudipeddi could correspond to an update request because “the back up when [an] old version of the file [is] overwritten with a newer version could essentially be considered an update request.” However, Applicant submits that recalling stored data does not teach updating data. One of skill in the art would not confuse the mere retrieval of stored data as teaching the updating of data. Assuming an updating of data is performed in some other computer system, there is no such disclosure in either Pudipeddi or Goudie. The Examiner’s reasoning is clearly a result of impermissible hindsight upon viewing the Applicant’s disclosure.

For at least the above reasons, claim 1 and its dependent claims should be deemed allowable. To the extent claims 5, 8, 11, and 15 recite similar subject matter, claims 5, 8, 11, and 15 and their dependent claims should be deemed allowable for at least the same reasons.

Claim 2

Claim 2 recites “wherein said order [of the data update requests] is an order in which said data update requests were made.” The Examiner asserts that paragraph 109 of Pudipeddi teaches this aspect of the claim. The aspect of Pudipeddi cited by the Examiner discloses that requests for data recall are queued sequentially and are processed in the order received. However, there is no teaching or suggestion that the data items 406 (cited for teaching the claimed data requests) are in an order in which the data requests were made. In particular, this would be contrary to the operation of Pudipeddi.

Pudipeddi is directed to making queues for each medium that contains a request. By the use of queues, data recall can be achieved more efficiently since the mounting and dismounting

of a medium is not required and shuttling back and forth on a same medium is not required. See paras. 49 and 50. Therefore, the purpose of Pudipeddi is to organize the requests for data.

Therefore, the order of the data requests in queues 402 are not in an order in which the data requests were made. More specifically, if the requests for data were made in the order in which the requests were made, then the system would have to shuttle back and forth between different locations based on the location of the desired information on the memory. See para. [0004] of Pudipeddi. This is clearly contrary to the operation of Pudipeddi.

Consequently, claim 2 should be deemed allowable. To the extent claims 6 and 9 recite similar subject matter, claims 6 and 9 should also be deemed allowable for at least the same reasons.

Claim 3

Claim 3 recites “said capacity corresponds to a number of said data update requests which said data processing unit is adapted to optimally process in order in said one block before processing said data update requests in said next one of said blocks.” The Examiner asserts that paragraph 40 of Goudie teaches this aspect of the claim. The paragraph of Goudie cited by the Examiner discloses:

The ‘empty blocks’ field is a byte wide field that indicates the number of spare (unused) blocks on the tail of a single queue, although some of the blocks may be used by an uncommitted write process which is still loaded). A single queue contains a maximum of 255 blocks, which enables the ‘empty blocks’ and ‘used blocks’ fields to be accessed together. The value in ‘empty blocks’ multiplied by the ‘block size’ gives a good approximation of the storage space (RAM) currently available, although this excludes any space available in the block which ‘committed tail block’ points to.

There is no teaching or suggestion that a capacity corresponds to a number of data update

requests which the data processing unit is adapted to optimally process in order in said one block before processing said data update requests in said next one of said blocks, as claimed.

Consequently, claim 3 should be deemed allowable. To the extent claims 7 and 10 recite similar subject matter, claims 7 and 10 should be deemed allowable for at least the same reasons.

Claim 17

Claim 17 recites “wherein the grouping of all of said data update requests into the plurality of blocks is performed at a same time.” As discussed above, Pupipeddi clearly discloses the addition of data requests at different points in time and the subsequent reordering of queues according to the time of the addition of a new data request. See Pudipeddi Figs. 8A-8B. Consequently, Pudipeddi does not teach the elements of claim 17 and claim 17 should be deemed allowable.

Claim 18

Claim 18 recites “wherein said blocks contain a predetermined number of the data update requests.” The Examiner asserts that Goudie para. [0023] teaches this aspect of the claim. The aspect of Goudie cited by the Examiner discloses grouped queues including elements queues and resource queues. Goudie is not at all concerned with data update requests. Consequently, Goudie does not teach the elements of claim 18 and claim 18 should be deemed allowable.

Claim 19

Claim 19 recites “wherein said blocks are grouped into a package according to a common key among the blocks.” The Examiner asserts that Goudie discloses grouping blocks according to a common size. Applicant submits that grouping blocks according to size (which the Examiner has also cited for teaching the claimed capacity) does not teach a common key. Further, one of skill in the art would not confuse the grouping of data based on size as teaching

grouping blocks according to a common key among the blocks. For at least the above reasons, claim 19 should be deemed allowable.

III. Rejection of claims 11-14 under 35 U.S.C. § 103

Claims 11-14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pudipeddi in view of Goudie and in further view of In re Harza, 274 F.2d 669, 671, 124 USPQ 378, 380 (CCPA 1960). To the extent claim 11 recites subject matter similar to claim 1, it should be deemed allowable for at least the same reasons. Further, dependent claims 12-14 should also be deemed allowable for at least the same reasons.

IV. Claim Rejections under 35 U.S.C. § 103

Claim 15 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Pudipeddi et al. (U.S. Pub. No. 2005/0165820). To the extent claim 15 recites subject matter similar to claim 1, it should be deemed allowable for at least the same reasons.

V. Conclusion

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

/Ruthleen E. Uy/

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Ruthleen E. Uy
Registration No. 51,361

Date: July 19, 2007